Coding : Jurnal Komputer dan Aplikasi
Volume 07, No. 01 (2019), hal. 85-96 ISSN: 2338-493X

PENERAPAN ALGORITMA *KNAPSACK* DAN FUNGSI *HASH* PADA SISTEM *E-VOTING*

(Studi Kasus: Pemilihan Raya Mahasiswa Universitas Tanjungpura Pontianak)

[1]Nircho Dwi Anggoro, [2]Cucu Suhery, [3]Ikhwan Ruslianto
[1] [2] [3]Jurusan Rekayasa Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jalan Prof. Dr. H. Hadari Nawawi, Pontianak
Telp./Fax.: (0561) 577963
e-mail : [1]nirchodwianggoro94@gmail.com [2]csuhery@siskom.untan.ac.id
[3]ikhwanruslianto@siskom.untan.ac.id

ABSTRAK

Electronic voting (e-voting) adalah suatu metode pemungutan dan perhitungan suara pada pemilihan kandidat menggunakan perangkat elektronik dimana data dicatat, disimpan dan diproses dalam bentuk informasi digital. Pada e-voting terdapat beberapa syarat keamanan yang harus dipenuhi yaitu kerahasiaan dan keaslian data hasil pemilihan. Pada penelitian ini dibuat sistem e-voting dengan menggunakan enkripsi algoritma Knapsack, fungsi Hash, dan encode base64 untuk memenuhi syarat keamanan pada e-voting. Berdasarkan hasil pengujian dengan melakukan perubahan data hasil pemilihan pada database sistem e-voting, diperoleh bahwa algoritma Knapsack dapat membuat data hasil pemilihan menjadi data acak yang tidak dapat dibaca sehingga tidak dapat diketahui oleh pihak yang tidak berwenang, dan fungsi Hash dapat mendeteksi terjadinya perubahan data pada hasil pemilihan, sehingga data hasil pemilihan dapat terjamin kerahasiaan dan keasliannya.

Kata Kunci: *E-voting, Keamanan E-voting, Knapsack, fungsi Hash*

1. PENDAHULUAN

Pemilihan Raya Mahasiswa atau disingkat Pemirama merupakan pelaksanaan pemilihan untuk menentukan pasangan Presiden dan Wakil Presiden Mahasiswa sebagai pemimpin pada Badan Eksekutif Mahasiswa (BEM) di setiap Perguruan Tinggi di Indonesia termasuk di Perguruan Tinggi Negeri Universitas Tanjungpura (Untan) Pontianak.

Pemirama Untan dilaksanakan secara konvensional dengan menggunakan kertas suara. Pelaksanaan pemilihan secara konvensional memiliki beberapa kekurangan seperti besarnya biaya yang digunakan untuk mencetak kertas suara, banyaknya kertas suara tidak sah pada saat pemilihan, proses perhitungan suara yang lambat dan rentan terjadi manipulasi pada data hasil pemilihan berdasarkan Berita Acara Pelaksanaan (BAP) Pemirama Untan tahun 2017. Permasalahan pada pemilihan konvensional dapat diatasi dengan menerapkan metode pemungutan suara menggunakan perangkat elektronik atau yang lebih dikenal dengan istilah *electronic voting* atau *e-voting.*

E-Voting adalah suatu metode pemungutan dan perhitungan suara pada pemilihan, dimana data dicatat, disimpan dan diproses dalam bentuk informasi digital. Pelaksanaan pemilihan menggunakan sistem *e-voting* memiliki syarat yang harus dipenuhi untuk menjaga aspek keamanan informasi dengan tujuan menjaga kerahasiaan dan keaslian data pemilihan[7].

Penelitian tentang keamanan informasi pada *e-voting* telah banyak dilakukan salah satunya oleh Rahmadian (2014)[5], pada penelitian tersebut

dilakukan proses untuk menjaga kerahasiaan data pemilihan atau dikenal dengan istilah *confidentiality* dengan menggunakan metode kriptografi algoritma kunci publik *Rivest Shamir Adleman* (RSA). Pengujian tersebut dilakukan dengan skenario manipulasi data yang dilakukan oleh admin utama. Hasil dari penelitian tersebut adalah algoritma kunci publik RSA dapat digunakan untuk menjaga kerahasiaan dan mengetahui keaslian data pemilihan.

Penelitian lain tentang *e-voting* dilakukan oleh Wisnu (2014)[10]. Pada penelitian tersebut digunakan proses *hashing* dan *digital signing* untuk menjaga kerahasiaan data hasil pemilihan.

Penelitian tentang *e-voting* juga dilakukan oleh Ridwan (2016)[6]. Pada penelitian tersebut telah dibuat sistem *e-voting* berbasis *website* dengan menerapkan metode keamanan algoritma kunci publik RSA untuk menjamin kerahasiaan data hasil pemilihan. Hasil penelitian disimpulkan bahwa algortima RSA dapat digunakan untuk menjamin kerahasiaan data hasil pemilihan pada sistem *e-voting*.

Kerahasiaan dan keaslian data pada sistem *e-voting* merupakan aspek penting yang harus dipenuhi. Oleh Karena itu dibutuhkan metode yang dapat menjaga kerahasiaan dan mengetahui keaslian data pemilihan pada sistem *e-voting*, salah satu cara yang dapat dilakukan adalah dengan menggunakan metode Kriptografi. Kriptografi adalah teknik untuk menjaga kerahasiaan dan keaslian data agar tidak dapat diketahui oleh pihak-pihak yang tidak berhak mengaksesnya.

Berdasarkan permasalahan tentang keamanan informasi pada *e-voting*, maka pada penelitian ini dilakukan proses untuk menjaga kerahasiaan dan keaslian data pemilihan dengan menggunakan metode kriptografi algoritma *Knapsack* dan fungsi *Hash*.

Knapsack adalah algoritma yang digunakan untuk melakukan proses pengubahan data yang dapat dibaca menjadi data acak yang tidak dapat dibaca sedangkan fungsi *Hash* digunakan untuk mendeteksi keaslian data pada sistem *e-voting*.

2. LANDASAN TEORI

2.1 *E-Voting*

E-Voting adalah suatu metode pemungutan dan perhitungan suara pada pemilihan menggunakan perangkat elektronik dimana data dicatat, disimpan dan diproses dalam bentuk informasi digital. Beberapa syarat keamanan pada sistem *e-voting* adalah sebagai berikut:

1. *Eligibility*: Hanya pemilih yang terdaftar yang dapat melakukan pemilihan.
2. *Unreusability*: Setiap pemilih hanya bisa memilih satu kali.
3. *Anonymity*: Pilihan pemilih dirahasiakan.
4. *Accuracy*: Pilihan pemilih tidak bisa dirubah atau dihapus selama atau setelah pemilihan dan juga tidak bisa ditambahkan setelah pemilihan ditutup.
5. *Fairness*: Perhitungan suara sebelum pemilihan ditutup tidak bisa dilakukan.
6. *Vote and Go*: Pemilih hanya dapat melakukan pemilihan saja.
7. *Public Verifiability*: Setiap orang dapat melihat hasil perolehan suara saat proses pemilihan selesai.

Untuk menjamin keamanan informasi pada sistem *evoting*, ada 3 aspek dasar yang harus dipenuhi yaitu:

1. Kerahasiaan (*Confidentiality*) merupakan aspek untuk menjaga kerahasiaan suatu informasi dari semua pihak yang tidak memiliki hak akses terhadap infomasi tersebut.
2. Integritas (*Integrity*) merupakan aspek untuk menjaga keaslian dan keakuratan informasi agar tidak terjadi perubahan oleh pihak yang tidak memiliki otoritas untuk merubah informasi tersebut. Untuk menjamin integritas data, harus memiliki kemampuan untuk mendeteksi perubahan informasi, seperti penyisipan, penghapusan dan penggantian.

3. Ketersediaan (*Availability*) merupakan aspek untuk menjamin ketersediaan informasi ketika di-butuhkan kapanpun tanpa adanya gangguan dan tidak dalam format yang tidak bisa digunakan.

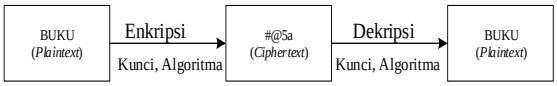
Tiga aspek dasar *confidentiality*, *integrity*, dan *availability* (CIA) merupakan dasar diantara program-program keamanan yang dikembangkan. Ketiga aspek tersebut merupakan aspek yang saling berhubungan dalam konsep keamanan informasi[8].

2.2 Kriptografi

Kriptografi berasal dari bahasa Yunani, yaitu *kripto* dan *graphia*. *Kripto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Kriptografi merupakan ilmu dan seni dalam memproteksikan informasi dengan mengubahnya ke dalam bentuk himpunan karakter acak yang tidak dapat dibaca. Kriptografi adalah sebuah cara yang efektif dalam mengamankan informasi-informasi penting baik yang tersimpan dalam media penyimpanan maupun yang ditransmisikan melalui jaringan komunikasi[2]. Pada prinsipnya, kriptogarfi memiliki 4 komponen utama yaitu:

1. *Plaintext*, yaitu pesan yang dapat dibaca.
2. *Ciphertext*, yaitu pesan yang disandikan.
3. *Key*, yaitu kunci untuk melakukan enkripsi atau dekripsi.
4. Algoritma, yaitu metode yang digunakan untuk melakukan enkripsi atau dekripsi.

Enkripsi adalah sebuah proses menjadikan *plaintext* menjadi *ciphertext*. Sedangkan dekripsi merupakan proses mengubah *ciphertext* menjadi *plaintext*. Proses enkripsi dan dekripsi menggunakan kunci dapat dilihat pada Gambar 1.



Gambar 1. Ilustrasi Enkripsi Dekripsi

2.3 Algoritma *Knapsack*

Algoritma *Knapsack* adalah algoritma kriptografi kunci publik. Keamanan algoritma ini terletak pada sulitnya memecahkan persoalan *Knapsack* (*Knapsack Problem*). *Knapsack* artinya karung atau kantung yang mempunyai kapasitas muat terbatas. Barang-barang dimasukkan kedalam karung hanya sampai batas kapasitas maksimum karung saja. Algoritma telah dimodifikasi menjadi *non-superincreasing Knapsack* dengan menggunakan kunci publik untuk enkripsi dan kunci pribadi untuk dekripsi. Kunci publik merupakan barisan *non-superincreasing* sedangkan kunci pribadi tetap merupakan barisan *superincreasing*. Modifikasi ini ditemukan oleh Martin Hellman dan Ralph Merkle. Pada algoritma *Knapsack*, kunci publik dibuat berdasarkan kunci pribadi yang ada, oleh karena itu dapat disimpulkan bahwa kunci pribadi dapat digunakan untuk membuat kunci publik dan mendekripsikan pesan yang diterima [2].

Tahapan Pembuatan Kunci Publik dan Kunci Pribadi *Knapsack* adalah sebagai berikut:

1. Tentukan barisan *superincreasing s* yang merupakan kunci pribadi. Barisan *s* adalah barisan dimana setiap nilai di dalam barisan tersebut lebih besar dari jumlah semua nilai sebelumnya, sehingga:
   $si = (1, 2, \dots, n)$ (2.1)
2. Tentukan p sebagai kunci pribadi yang merupakan bilangan prima yang nilainya lebih besar dari jumlah *s*, sehingga:
   $p = \sum_{i=1}^{n} s_i$ (2.2)
3. Tentukan a sebarang bilangan bulat yang merupakan kunci pribadi dimana nilainya diantara 1 sampai p-1, sehingga:
   $1 \leq a \leq p - 1$ (2.3)
4. Tentukan t kunci publik dengan cara mengalikan nilai *a* dengan setiap elemen di dalam barisan *superincreasing s* modulus *p*, sehingga:
5. $ti = a * si \bmod p$ (2.4)

Dari proses pembuatan kunci diatas didapatkan kunci pribadi (*s, p, a*) dan kunci publik (*t*)

2.4 *Hash*

Fungsi *Hash* yang juga sering disebut fungsi *Hash* satu arah (*One-Way Function*), *message digest*, *fingerprint*, fungsi kompresi, dan *message authentication code* (MAC) merupakan suatu fungsi matematika yang mengambil masukan panjang variabel dan mengubahnya ke dalam urutan biner dengan panjang yang tetap. Masukan fungsi *Hash* adalah blok pesan (M) dan keluaran dari *Hashing* blok pesan sebelumnya[3]. Sehingga:

1. hi = H (Mi, hi – 1) (2.5)

2.5 Algortima *Base*64

Transformasi *Base64* merupakan salah satu algoritma untuk *encoding* dan *decoding* suatu data ke dalam *format* ASCII, yang didasarkan pada bilangan dasar 64 atau bisa dikatakan sebagai salah satu metode yang digunakan untuk melakukan *encoding* (penyandian) terhadap data *binary*. Karakter yang dihasilkan pada transformasi *Base64* ini terdiri dari A…Z, a….z dan 0….9, serta ditambah dengan dua karakter terakhir yang bersimbol yaitu + dan / serta satu buah karakter sama dengan (=) yang digunakan untuk penyesuaian dan menggenapkan data *binary* atau istilahnya disebut sebagai pengisi *pad*. Karakter simbol yang akan dihasilkan akan tergantung dari proses algoritma yang berjalan[9].

3. METODE PENELITIAN

Metode penelitian dimulai dari studi literatur yang merupakan penelusuran penelitian relevan yang telah dilakukan sebelumnya dan penelusuran dasar teori yang bertujuan untuk menunjang kebutuhan dalam penelitian.

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan dari sebuah perangkat lunak yang akan di bangun.

Perancangan sistem terdiri dari: perancangan sistem secara umum, perancangan enkripsi dan dekripsi menggunakan kriptografi algoritma *Knapsack.* perancangan basis data pada sistem. Perancangan antarmuka aplikasi.

Implementasi dilakukan setelah tahap perancangan selesai, dalam implementasinya, sistem dibuat menggunakan bahasa pemrograman PHP dan *database* MySQL.

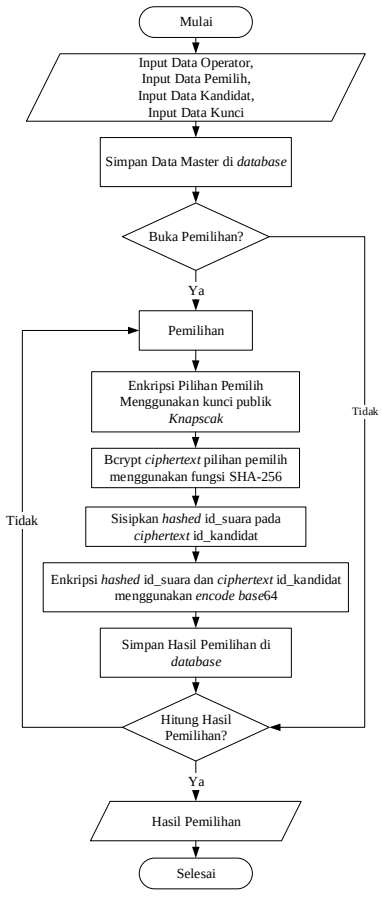
Pengujian sistem dilakukan untuk memastikan apakah sistem dapat bekerja dengan baik sesuai dengan perancangan yang telah dibuat.

4. PERANCANGAN

Perancangan sistem *e-voting* digambarkan menggunakan diagram alir atau *Flowchart*. Desain perancangan ini digunakan untuk membantu memahami alur kerja sistem secara keseluruhan.

4.1 Perancangan Sistem *E-Voting*

4.1.1 *Flowchart* Keseluruhan Sistem *E-Voting*



Gambar 2. *Flowchart* Keseluruhan Sistem *E-Voting*

Gambar 2 merupakan *Flowchart* keseluruhan sistem *e-voting*. Proses dimulai dengan memasukkan data operator, data pemilih tetap, data kandidat dan data kunci *Knapsack*. Setelah dimasukan, data tersebut disimpan pada *database*.

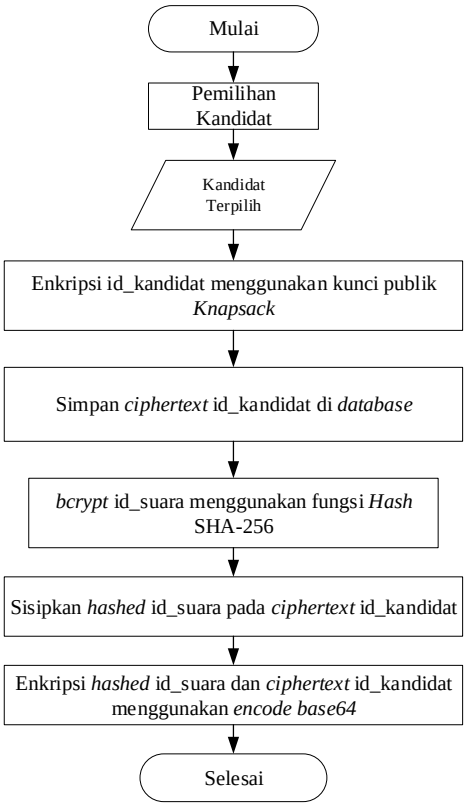
Ketika data telah disimpan di-*database* maka proses berikutnya adalah menentukan apakah proses pemilihan dibuka atau ditutup.

Jika proses pemilihan dibuka, maka akan dilakukan pemilihan dan hasil pemilihan tersebut disimpan di *database* kemudian akan dilakukan penghitungan hasil pemilihan.

Jika proses pemilihan ditutup, maka akan langsung dilakukan proses peng-hitungan hasil pemilihan. Jika dilakukan penghitungan hasil pemilihan, maka didapat hasil pemilihan dan proses selesai.

Jika penghitungan hasil pemilihan tidak dilakukan, maka kembali ke proses pemilihan.

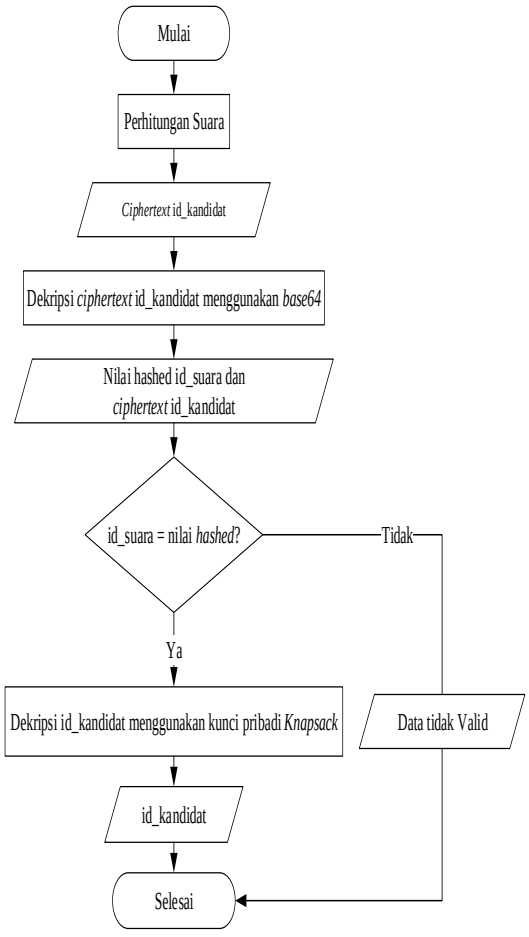
4.1.2 *Flowchart* Pemilihan



Gambar 3. *Flowchart* Pemilihan

Gambar 3 merupakan *flowchart* pemilihan yang dilakukan oleh pemilih menggunakan kunci publik *Knapsack*. Setelah dilakukan proses pemilihan dan didapatkan kandidat terpilih, selanjutnya sistem akan mengenkripsi id_kandidat terpilih menggunakan kunci publik *Knapsack.* Kemudian data yang telah dienkripsi disimpan pada *database*, setelah itu id_kandidat disisipkan nilai *hashed* dari id_suara kemudian hasilnya di enkripsi menggunakan *encode base*64.

4.1.3 *Flowchart* Perhitungan Hasil Pemilihan



Gambar 4. *Flowchart* Perhitungan Hasil Pemilihan

Gambar 4 merupakan *flowchart* perhitungan hasil pemilihan. Proses perhitungan dimulai dengan mendekripsi *ciphertext* id_kandidat menggunakan

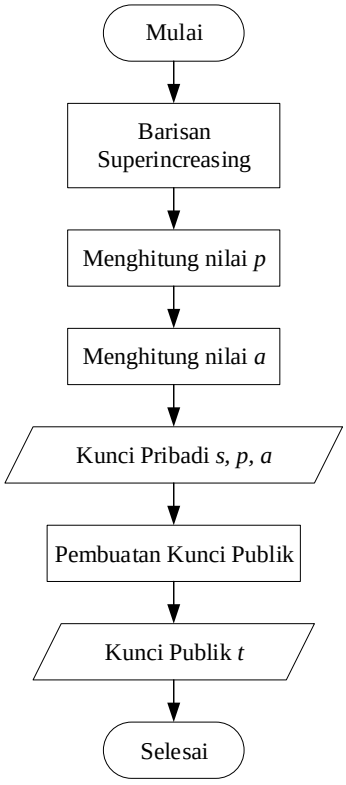
base64 terlebih dahulu, kemudian dari hasil dekripsi didapat nilai *hashed* id_suara dan *ciphertext* id_kandidat, setelah itu nilai *hashed* id_suara dibandingkan, apabila nilainya tidak sama maka data dinyatakan tidak valid atau telah terjadi perubahan, namun apabila id_suara sama dengan nilai *hashed* maka sistem akan mendekripsi id_kandidat menggunakan kunci pribadi *Knapscak*.

Apabila kunci pribadi valid maka data hasil pemilihan dapat di dekripsi dan didapatkan *plaintext* id_kandidat hasil pemilihan. Setelah didapatkan data hasil pemilihan maka dilakukan proses perhitungan untuk mendapatkan data total pemilihan dan proses selesai.

4.2 Perancangan Algoritma *Knapsack*

4.2.1 Pembuatan Kunci *Knapsack*

Pembuatan Kunci *Knapsack* merupakan proses pembuatan kunci pribadi dan kunci publik *Knapsack*. Proses pembuatan kunci *Knapsack* dapat dilihat pada Gambar 5.



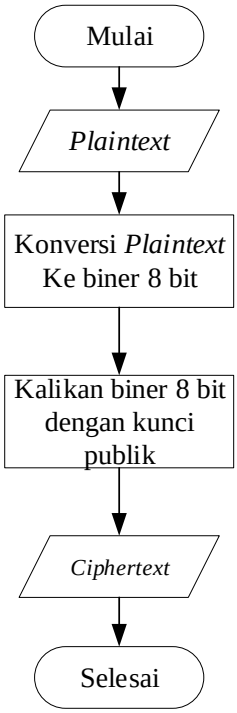
Gambar 5. *Flowchart* Pembuatan Kunci *Knapsack*

Gambar 5 merupakan *flowchart* pembuatan kunci pribadi dan kunci publik *Knapsack*. Pembuatan kunci pribadi dimulai dengan proses pembuatan barisan *superincreasing* oleh sistem. Kemudian sistem menghitung nilai *p* dan nilai *a*. Setelah itu didapat kunci pribadi *knapsack* yaitu nilai *s*, *p* dan *a*.

Kemudian untuk mendapatkan kunci publik maka dilakukan dengan cara mengalikan nilai *a* dengan setiap elemen di dalam barisan *superincreasing s* modulus *p*. Setelah kunci publik didapat maka proses pembuatan kunci selesai.

4.1.2 Proses Enkripsi

Enkripsi adalah proses menjadikan *plaintext* menjadi *ciphertext*. Perancangan proses enkripsi menggunakan algoritma *Knapsack* dapat dilihat pada Gambar 6.



Gambar 6. *Flowchart* Enkripsi

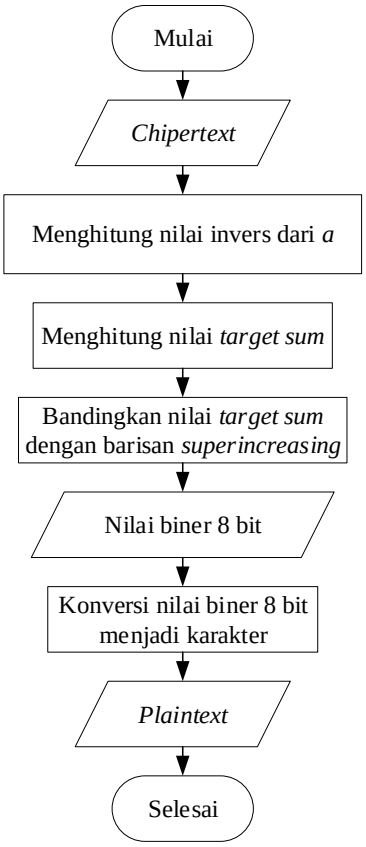
Gambar 6 merupakan *flowchart* enkripsi data pemilihan menggunakan kunci publik *Knapsack*. Proses enkripsi dimulai dengan masukan data *plaintext* yang didapatkan dari proses pemilihan,

kemudian dilakukan proses konversi data *plaintext* ke biner 8-bit.

Selanjutnya setiap bit biner dikalikan dengan kunci publik *Knapsack* dan hasil perkalian dijumlahkan untuk mendapatkan data *ciphertext*. Setelah data *ciphertext* didapatkan maka proses enkripsi selesai.

4.1.3 Proses Dekripsi

Proses dekripsi adalah proses mengubah *ciphertext* menjadi *plaintext*. Proses dekripsi menggunakan algoritma *Knapsack* dapat dilihat pada Gambar 7.

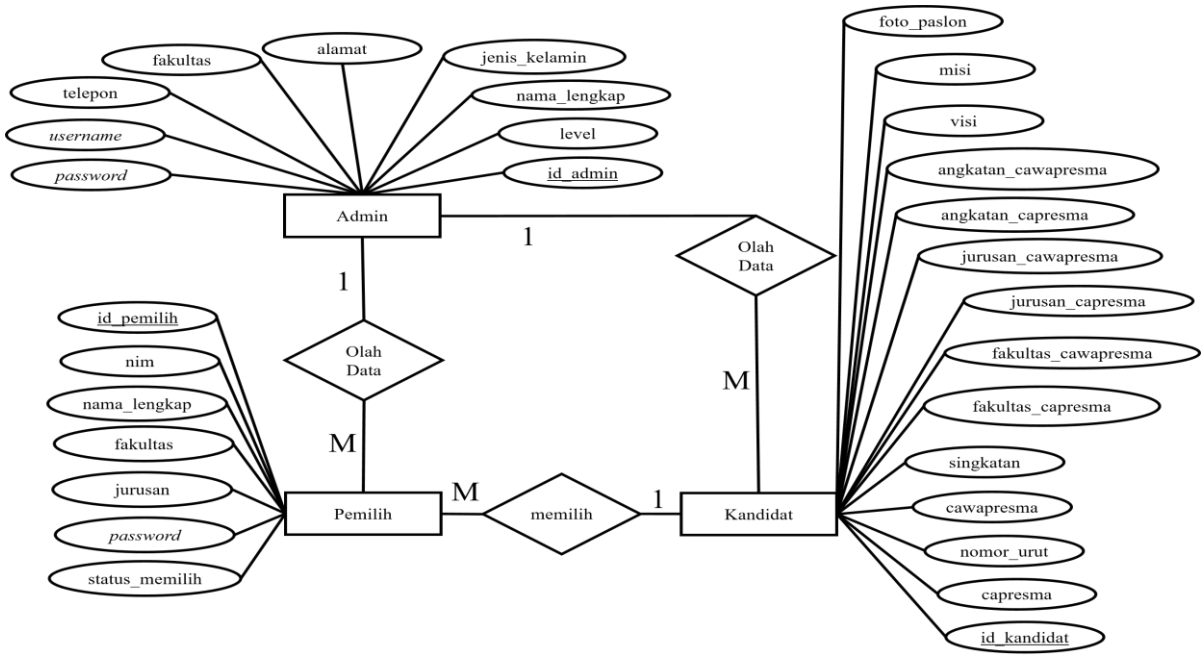


Gambar 7. *Flowchart* Dekripsi

Gambar 7 merupakan *flowchart* dekripsi. Proses dekripsi dimulai dengan masukan data *ciphertext*, kemudian dilakukan proses hitung nilai *invers* dari *a* yang merupakan kunci pribadi, selanjutnya dilakukan proses hitung *target sum*. Bandingkan nilai *target sum* dengan barisan *superincreasing s* yang merupakan kunci pribadi untuk mendapatkan nilai biner 8-bit. Kemudian nilai biner 8-bit dikonversi menjadi karakter untuk mendapatkan *plaintext*. Setelah *plaintext* didapatkan maka proses selesai.

4.2 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) berguna untuk membantu meng-organisasikan data dalam suatu sistem ke dalam entitas-entitas dan menentukan hubungan antar entitas. Perancangan ERD sistem *e-voting* dapat dilihat pada Gambar 8.

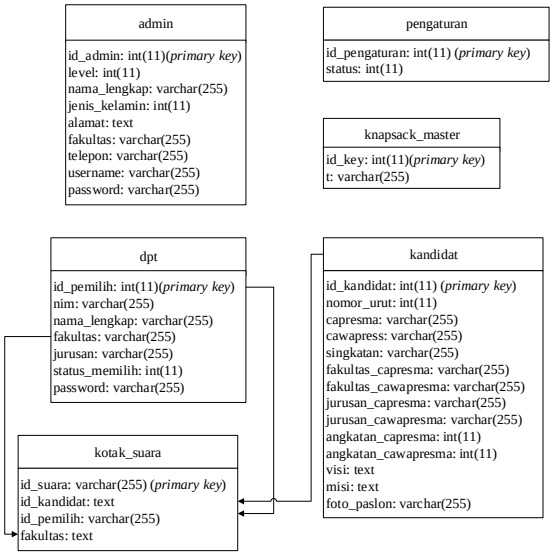


Gambar 8. *Entity Relationship Diagram* (ERD)

Gambar 8 adalah ERD dari sistem *e-voting* yang memiliki 3 entitas dengan 3 relasi, dengan derajat kardinalitas *many to one* untuk entitas pemilih memilih kandidat dan derajat kardinalitas *one to many* untuk entitas admin olah data kandidat dan pemilih.

4.3 Relasi Tabel Basis Data

Relasi antar tabel pemilihan pada sistem *e-voting* dapat dilihat pada Gambar 9.

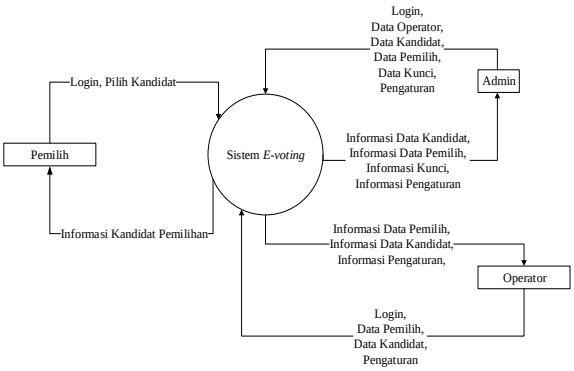


Gambar 9. Relasi Tabel *E-Voting*

Gambar 9 merupakan relasi antara tabel ‘kandidat’, tabel ‘dpt’, dan tabel ‘kotak_suara’. Tabel ‘kandidat’ digunakan untuk menyimpan data kandidat pemilihan. Tabel ‘dpt’ digunakan untuk menyimpan data pemilih pada proses pemilihan. Tabel ‘kotak_suara’ digunakan untuk menyimpan data hasil pemilihan. Ada beberapa atribut pada tabel ‘dpt’ dan tabel ‘kandidat’ yang nantinya akan digunakan pada tabel ‘kotak_suara’ yaitu atribut id_kandidat, id_pemilih, dan fakultas.

4.4 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) berfungsi untuk menggambarkan proses aliran data yang terjadi pada sistem *e-voting*. Diagram konteks direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. Diagram konteks *e-voting* dapat dilihat pada Gambar 10.

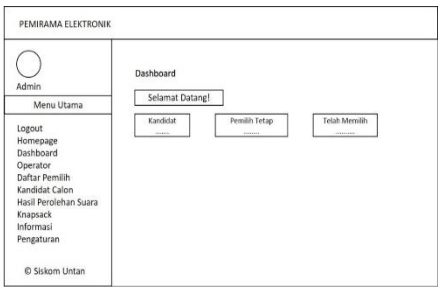


Gambar 10. Diagram Konteks *E-Voting*

Gambar 10 adalah diagram konteks yang memberikan gambaran umum bahwa sistem berinteraksi dengan tiga entitas yaitu pemilih, operator, dan admin.

4.5 Perancangan Antar Muka (*interface*)

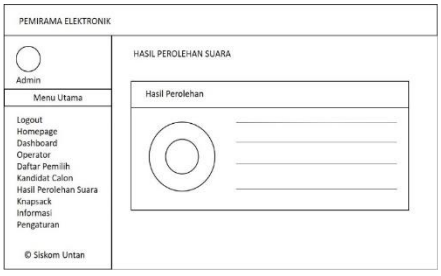
1. Perancangan Halaman Menu Utama *Admin* dapat dilihat pada Gambar 11.



Gambar 11. Perancangan Halaman Menu Utama Admin

2. Perancangan Halaman Lihat Hasil Pemilihan

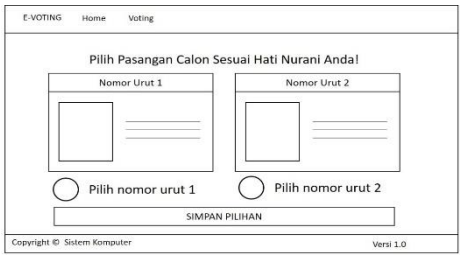
Halaman ini berfungsi untuk menampilkan data hasil perolehan suara. Perancangan halaman lihat hasil pemilihan dapat dilihat pada Gambar 12.



Gambar 12. Perancangan Halaman Lihat Hasil Perolehan Suara

3. Perancangan Halaman Pemilihan

Halaman ini berfungsi untuk melakukan proses pemilihan kandidat. Perancangan lihat hasil perolehan suara dapat dilihat pada Gambar 13.



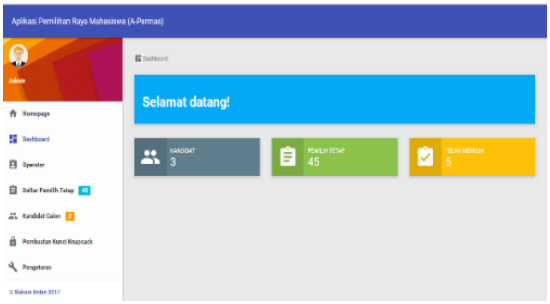
Gambar 13. Perancangan Halaman Pemilihan

5. IMPLEMENTASI, PENGUJIAN DAN PEMBAHASAN

5.1 Implementasi Antarmuka

1. Tampilan Halaman Utama Admin

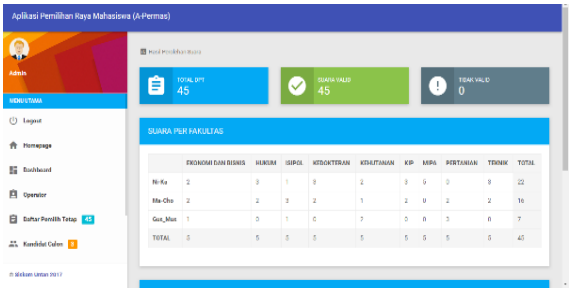
Halaman utama admin adalah halaman untuk mengolah olah data operator, daftar pemilih tetap, kandidat calon, pembuatan kunci *Knapsack*, lihat hasil perolehan suara, pengaturan aplikasi, pengaturan akun admin dan menu *logout*. Tampilan Halaman utama admin dapat dilihat pada Gambar 14.



Gambar 14. Tampilan Utama Halaman Admin

2. Tampilan Halaman Lihat Hasil Pemilihan

Halaman hasil pemilihan berfungsi untuk menampilkan hasil perolehan suara setelah pemilihan selesai dilaksanakan. Tampilan Halaman Lihat Hasil Pemilihan dapat dilihat pada Gambar 15.



Gambar 15. Tampilan Halaman Lihat Hasil Pemilihan

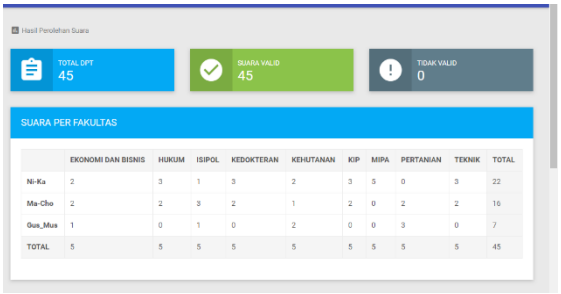
3. Tampilan Halaman pemilihan

Halaman pemilihan adalah halaman yang menampilkan data kandidat pemilihan raya mahasiswa. Pada halaman pemilihan pemilih dapat memberikan suaranya pada kandidat calon yang ada. Tampilan halaman pemilihan dapat dilihat pada Gambar 16.

Gambar 16. Tampilan Halaman Pemilihan

5.2 Pengujian Pemilihan

Pengujian pemilihan dilakukan dengan simulasi pemilih melakukan pemilihan kandidat dengan memilih nomor urut kandidat. Hasil pemilihan dapat dilihat pada Gambar 17.

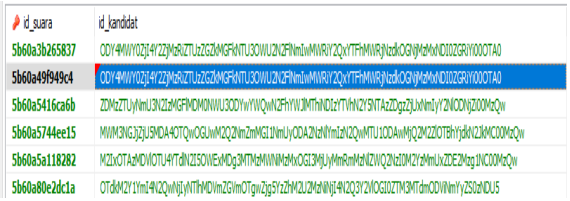


Gambar 17. Hasil Pemilihan

Gambar 17 merupakan hasil pemilihan pada sistem *e-voting*. Total pemilih yang terdaftar pada sistem *e-voting* adalah 45 pemilih dengan rincian 5 pemilih pada Fakultas Hukum, 5 pemilih pada Fakultas Pertanian, 5 pemilih pada Fakultas Ekonomi dan Bisnis, 5 pemilih pada Fakultas Teknik, 5 pemilih pada Fakultas Isipol, 5 pemilih pada Fakultas KIP, 5 pemilih pada Fakultas Kehutanan, 5 pemilih pada Fakultas MIPA dan 5 pemilih pada Fakultas Kedokteran. Rincian perolehan suara pada kandidat pemilihan yaitu Nomor Urut 1 memperoleh 22 suara, Nomor Urut 2 memperoleh 16 suara dan Nomor Urut 3 memperoleh 7 suara. Hasil pemilihan pada sistem *e-voting* semuanya dinyatakan valid karena tidak ada perubahan atau manipulasi pada data pemilihan.

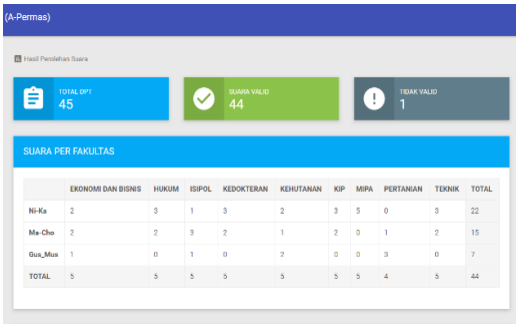
5.3 Pengujian Manipulasi Data

Pengujian manipulasi data dilakukan dengan cara mengubah id_kandidat pada tabel kotak suara pada database. Pengujian manipulasi data dapat dilihat pada Gambar 18.



Gambar 18. Manipulasi Data Pemilihan

Gambar 18 merupakan manipulasi data yang dilakukan dengan cara menggandakan id_kandidat. Setelah id_kandidat digandakan maka hasil pemilihan akan dinyatakan tidak valid oleh sistem. Hasil validasi pada sistem dapat dilihat pada Gambar 19.



Gambar 19. Hasil Validasi Data Pemilihan

Gambar 19 adalah tampilan hasil pemilihan pada sistem *e-voting*, setelah data pemilihan dimanipulasi pada *database*, maka sistem akan memberi informasi ada data tidak valid dan menyebabkan total hasil pemilihan berkurang, proses deteksi perubahan data pada hasil pemilihan dilakukan menggunakan fungsi *Hash*.

5.4 Pembahasan

Berdasarkan hasil pengujian untuk menjaga kerahasiaan dan keaslian data pada sistem *e-voting* dilakukan dengan menggunakan metode kriptografi algoritma *Knapsack*.

Data hasil pemilihan dienkripsi menggunakan kunci publik *Knapsack* kemudian data disimpan pada database, id_suara pada database di *bcrypt* menggunakan fungsi *Hash 256*, selanjutnya *hashed* id_suara disisipkan pada *ciphertext* id_kandidat, setelah itu dienkripsi menggunakan *base base*64.

Proses untuk melakukan perhitungan hasil pemilihan dimulai dengan mendekripsi *ciphertext* id_kandidat menggunakan *base64* terlebih dahulu, kemudian dari hasil dekripsi didapat nilai *hashed* id_suara dan *ciphertext* id_kandidat, setelah itu nilai *hashed* id_suara dibandingkan, jika nilainya tidak sama maka data dinyatakan tidak valid atau telah terjadi perubahan, jika id_suara sama dengan nilai *hashed* maka sistem akan mendekripsi id_kandidat menggunakan kunci pribadi *Knapscak*.

Jika kunci pribadi valid maka data hasil pemilihan dapat di dekripsi dan didapatkan *plaintext* id_kandidat hasil pemilihan. Setelah didapatkan data hasil pemilihan maka dilakukan proses perhitungan untuk mendapatkan data total pemilihan dan proses selesai.

6. PENUTUP

6.1 Kesimpulan

Berdasarkan hasil implementasi dan pengujian sistem *e-voting* dapat diambil kesimpulan sebagai berikut:

1. Algoritma *Knapsack* dapat menjaga kerahasiaan data hasil pemilihan dengan cara mengenkripsi id_kandidat terpilih menggunakan kunci publik *Knapsack* menjadi data acak yang tidak dapat dibaca, sehingga data hasil pemilihan tidak bisa diketahui oleh pihak yang tidak berhak mengetahuinya.
2. Fungsi hash dapat menjaga keaslian data hasil pemilihan dengan cara, id_suara di *bcrypt* menggunakan fungsi *Hash*256 menggunakan *salt*, sehingga data hasil pemilihan dapat dideteksi apabila terjadi perubahan atau manipulasi data.

6.2 Saran

Berdasarkan hasil penelitian ini terdapat beberapa saran yang dapat dijadikan penelitian lebih lanjut, antara lain:

1. Pada penelitian selanjutnya proses enkripsi untuk menjaga kerahasiaan dan keaslian data pada pemilihan dapat dilakukan dengan menggunakan algoritma lain sebagai perbandingan.
2. Pada penelitian selanjutnya sistem *e-voting* dapat diterapkan pada kegiatan-kegiatan pemilihan lainnya.

7. DAFTAR PUSTAKA

[1] Angga, C. (2011). Analisis Cara Kerja Beragam Fungsi *Hash* Yang Ada.

[2] Ariyus, D. (2008). Pengantar Ilmu Kriptografi: Teori, Analisa, dan Implementasi. Yogyakarta: Andi.

[3] Isnaini, M. F. (2013). Penerapan Sistem *E-Voting* pada Pemilihan Kepala Daerah di Indonesia (*The Application of E-Voting Systems in the Local Election in Indonesia*).

[4] Istiqomah, N. (2016). Sistem Keamanan *E-Voting* Menggunakan Fungsi *Hash* dan Algoritma *One Time Pad*.

[5] Rahmadian, T. (2014). Desain dan Implementasi Keamanan Sistem *E-Voting* dengan Jaminan *Confidentiality* Data.

[6] Ridwan, M. (2016). Rancang Bangun *E-Voting* dengan menggunakan Keamanan Algoritma Rivest Shamir Adleman (RSA) Berbasis Web (Studi Kasus: Pemilihan Ketua BEM FMIPA).

[7] Rokhman, A. (2011). Prospek dan Tantangan Penerapan *E-Voting* di Indonesia.

[8] Stenbro, M. (2010). *A Survey of Modern Electronic Voting*.

[9] Teguh, S. (2007). Pemanfaatan MIME *Base*64 untuk menyembunyikan *Source Code* PHP.

[10] Wisnu, D. A. (2014). Rancang Bangun Sistem *E-Voting* dengan menerapkan *Hash* dan *Digital Signature* untuk Verfikasi Data Hasil *Voting*.